New Golang Ransomware Agenda Customizes Attacks

: 8/25/2022

Ransomware

A new piece of ransomware written in the Go language has been targeting healthcare and education enterprises in Asia and Africa. This ransomware is called Agenda and is customized per victim.

By: Mohamed Fahmy, Nathaniel Gregory Ragasa, Earle Maui Earnshaw, Bahaa Yamany, Jeffrey Francis Bonaobra, Jay Yaneza August 25, 2022 Read time: 7 min (1930 words)

We recently discovered a new piece of targeted ransomware that was created in the Go programming language and that explicitly targeted one of our customers. This was evidenced by the specific email addresses and credentials the ransomware used. Malware written in the Go language (aka Golang) has become common among threat actors. One possible reason for this uptick in popularity is that Go statically compiles necessary libraries, making security analysis much harder.

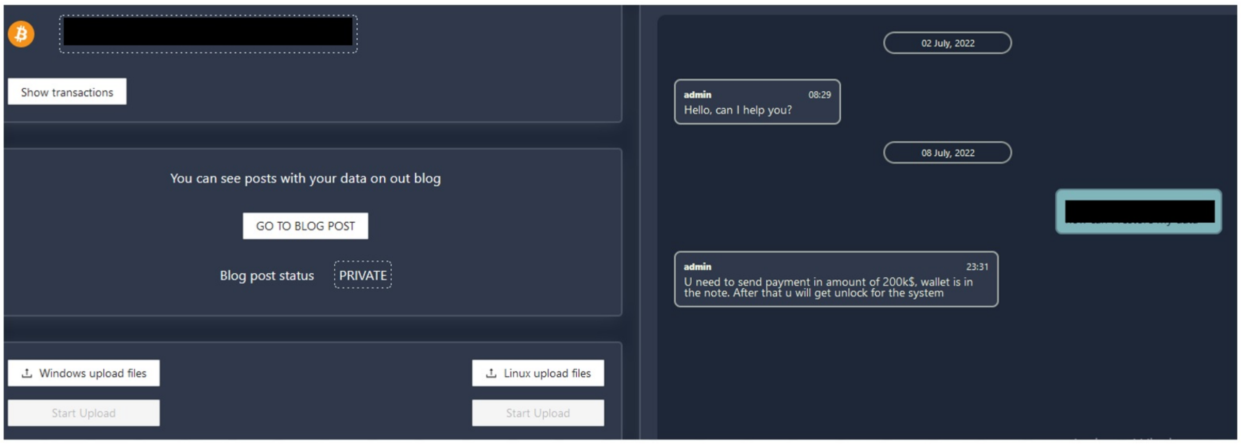
Our investigation revealed that the new ransomware in question targeted enterprises in Asia and Africa. Based on dark web posts by a user named “Qilin” (who seems to be connected to the ransomware distributors) and through ransom notes, the ransomware is called “Agenda.”

Agenda can reboot systems in safe mode, attempts to stop many server-specific processes and services, and has multiple modes to run. The samples of the ransomware that we collected were customized for each victim, and they included unique company IDs and leaked account details.

Targets

All collected samples were 64-bit Windows PE (Portable Executable) files written in Go, and they were aimed at Windows-based systems. The group distributing the malware was targeting healthcare and education organizations in Indonesia, Saudi Arabia, South Africa, and Thailand. Every ransomware sample was customized for the intended victim. Our investigation showed that the samples had leaked accounts, customer passwords, and unique company IDs used as extensions of encrypted files.

We believe that Qilin (or the Agenda ransomware group) offers affiliates options to customize configurable binary payloads for each victim, including details such as company ID, RSA key, and processes and services to kill before the data encryption. Also, the ransom amount requested is different per company, ranging from US$50,000 to US$800,000.



[redacted]

You can see posts with your data on out blog

Blog post status PRIVATE

02 July, 2022

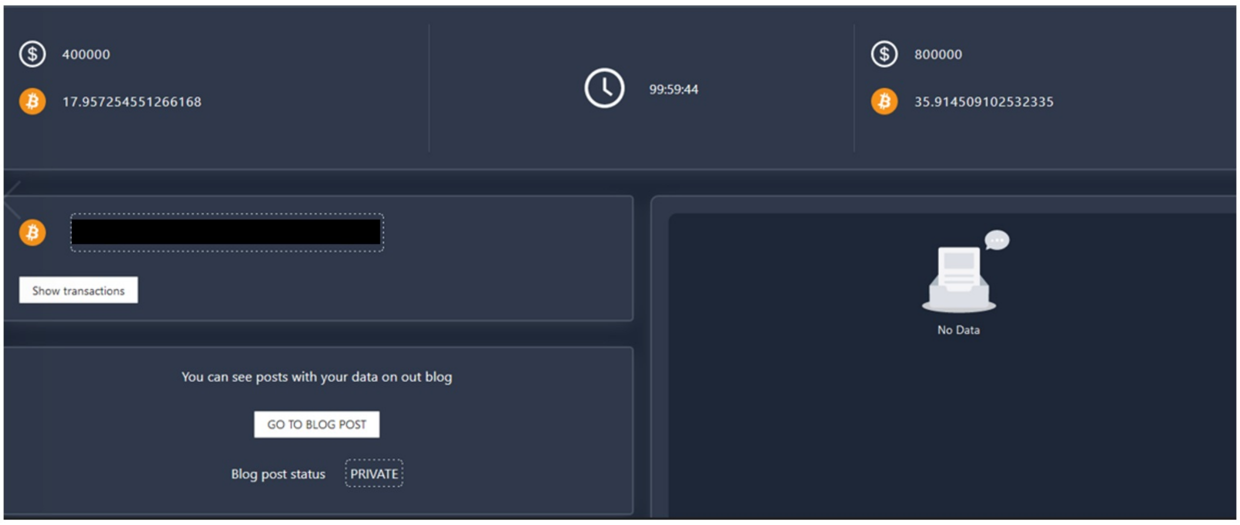
admin 08:29
Hello, can I help you?

08 July, 2022

[redacted]

admin 23:31
U need to send payment in amount of 200k$, wallet is in the note. After that u will get unlock for the system

Figure 1. An example of Qilin's ransom negotiations



400000
17.957254551266168

99:59:44

800000
35.914509102532335

[redacted]

You can see posts with your data on out blog

Blog post status PRIVATE

Figure 2. An example of ransom amount requested by Qilin

Similarities with other ransomware

We noticed some similarities between Agenda and the Black Basta, Black Matter, and REvil (aka Sodinokibi) ransomware.

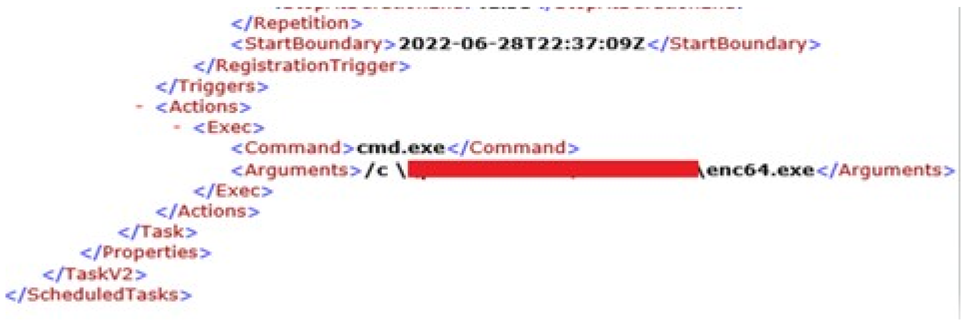
In terms of payment sites and the implementation of user verification on a Tor site, Agenda is very similar to Black Basta and Black Matter. Meanwhile, Agenda shares with Black Basta and REvil the same functionality of changing Windows passwords and rebooting in safe mode using this command:

C:\windows\system32\bcdedit.exe /set safeboot{current} network

Observed kill chain

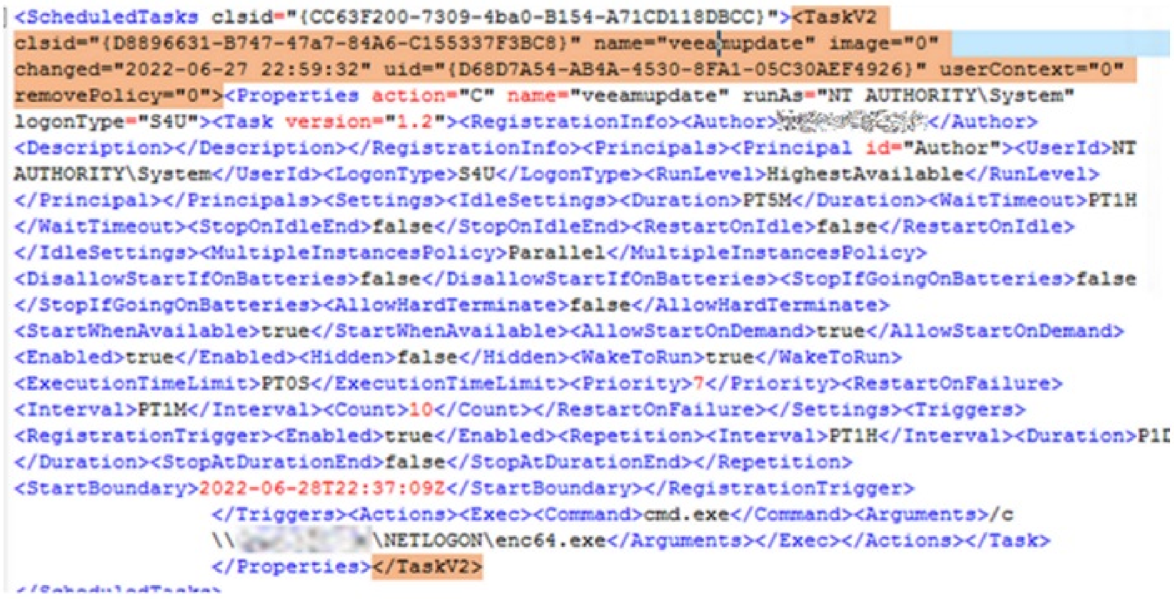
Investigating one incident involving this ransomware, we saw that the threat actor behind it used a public-facing Citrix server as a point of entry. We believe that the threat actor used a valid account to access this server and later move inside the victim's network. This was expected since the actor configured the ransomware with valid and privileged accounts.

The threat actor used RDP on Active Directory using leaked accounts. The actor dropped scanning tools, *Nmap.exe* and *Nping.exe*, for scanning the network. Next, the scheduled task was pushed by the group policy domain machine.



```
          </Repetition>
          <StartBoundary>2022-06-28T22:37:09Z</StartBoundary>
        </RegistrationTrigger>
      </Triggers>
    - <Actions>
      - <Exec>
          <Command>cmd.exe</Command>
          <Arguments>/c \[redacted]\enc64.exe</Arguments>
        </Exec>
      </Actions>
    </Task>
  </Properties>
</TaskV2>
</ScheduledTasks>
```

Figure 3. The scheduled task pushed by the group policy



```
<ScheduledTasks clsid="{CC63F200-7309-4ba0-B154-A71CD118DBCC}"><TaskV2
clsid="{D8896631-B747-47a7-84A6-C155337F3BC8}" name="veeamupdate" image="0"
changed="2022-06-27 22:59:32" uid="{D68D7A54-AB4A-4530-8FA1-05C30AEF4926}" userContext="0"
removePolicy="0"><Properties action="C" name="veeamupdate" runAs="NT AUTHORITY\System"
logonType="S4U"><Task version="1.2"><RegistrationInfo><Author>[redacted]</Author>
<Description></Description></RegistrationInfo><Principals><Principal id="Author"><UserId>NT
AUTHORITY\System</UserId><LogonType>S4U</LogonType><RunLevel>HighestAvailable</RunLevel>
</Principal></Principals><Settings><IdleSettings><Duration>PT5M</Duration><WaitTimeout>PT1H
</WaitTimeout><StopOnIdleEnd>false</StopOnIdleEnd><RestartOnIdle>false</RestartOnIdle>
</IdleSettings><MultipleInstancesPolicy>Parallel</MultipleInstancesPolicy>
<DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries><StopIfGoingOnBatteries>false
</StopIfGoingOnBatteries><AllowHardTerminate>false</AllowHardTerminate>
<StartWhenAvailable>true</StartWhenAvailable><AllowStartOnDemand>true</AllowStartOnDemand>
<Enabled>true</Enabled><Hidden>false</Hidden><WakeToRun>true</WakeToRun>
<ExecutionTimeLimit>PT0S</ExecutionTimeLimit><Priority>7</Priority><RestartOnFailure>
<Interval>PT1M</Interval><Count>10</Count></RestartOnFailure></Settings><Triggers>
<RegistrationTrigger><Enabled>true</Enabled><Repetition><Interval>PT1H</Interval><Duration>P1D
</Duration><StopAtDurationEnd>false</StopAtDurationEnd></Repetition>
<StartBoundary>2022-06-28T22:37:09Z</StartBoundary></RegistrationTrigger>
                </Triggers><Actions><Exec><Command>cmd.exe</Command><Arguments>/c
                \\[redacted]\NETLOGON\enc64.exe</Arguments></Exec></Actions></Task>
                </Properties></TaskV2>
```

Figure 4. The scheduled task created on the machine

We observed that there was only a short period between accessing the Citrix server and the ransomware infection: less than two days. The threat actor seemed to have scanned the network on the first day, and then a Group Policy Object (GPO) was created and the ransomware was deployed on the machines.

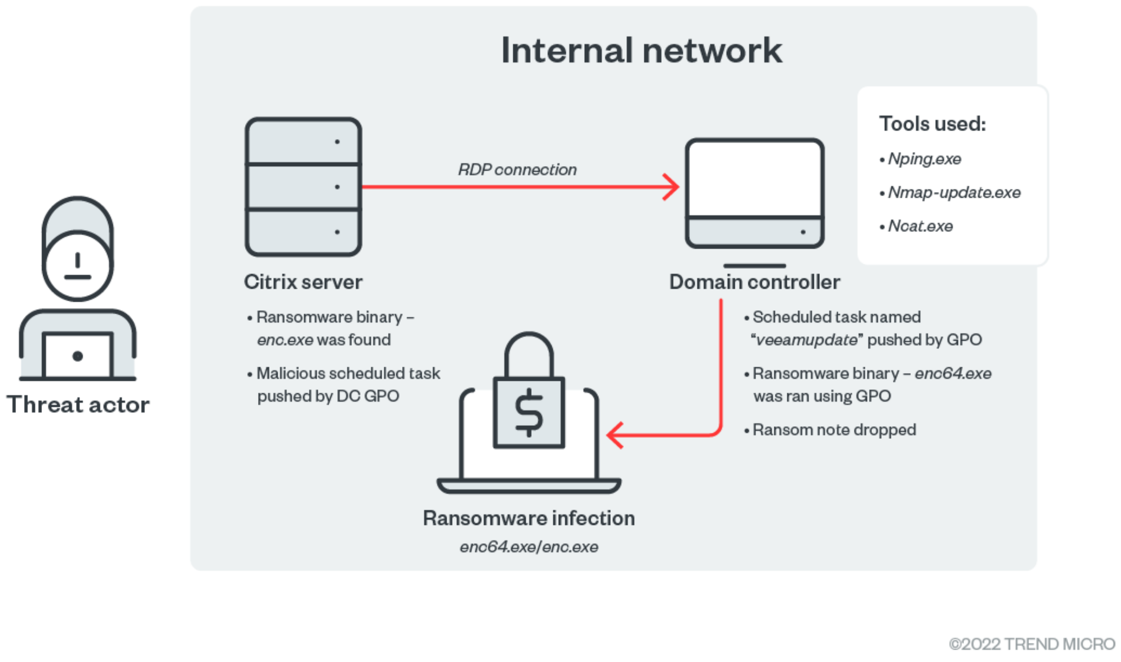


Figure 5. The Agenda ransomware's kill chain

Analysis and notable features

The Agenda ransomware is a 64-bit Windows PE file written in Go. Go programs are cross-platform and completely standalone, meaning they will execute properly even without a Go interpreter installed on a system. This is possible since Go statically compiles necessary libraries (packages).

Upon execution, this ransomware accepts various command-line arguments that define the malware flow and functionality, as listed in the table below.

Argument	**Description**
-alter {int}	Defines the port number for this child process
-encryption {value}	Allows for redefining the embed encryptor config to the customized choice
-ips {IP Address}	Allows for providing IP addresses
-min-size {value}	Defines the minimum file size to encrypt (e.g., 1 KB, 1 MB, 1 GB, 666 KB)
-no-proc	Defines the processes that will not be killed
-no-services	Defines the services that will not be killed
-password {string}	Defines the password to enter landing
-path {directory}	Defines the path that parses directories; if this flag is used and left empty, all directories will be scanned
-safe	Boots in safe mode
-stat	Makes malware print its configuration (processes and services to be killed, encryption, etc.)

Table 1. Command-line arguments accepted by Agenda

Agenda builds a runtime configuration to define its behavior, including its public RSA key, encryption conditions, list of processes and services to terminate, encryption extension, login credentials, and ransom note.

Runtime configuration component	Description
public_rsa_pem	RSA public key
directory_black_list	Directories excluded from encryption
file_black_list	File names excluded from encryption
file_pattern_black_list	File name extensions excluded from encryption
process_black_list	Processes to terminate
win_services_black_list	Services to terminate
company_id	Encryption extension
accounts	Login credentials
note	Ransom note

Table 2. The runtime configuration components of Agenda

As part of its initial routine, Agenda determines if the machine is running in safe mode by checking the string safeboot in the data of this registry value:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control SystemStartOptions

If it detects that the machine is running in safe mode, it terminates execution.

The ransomware then removes shadow volume copies via execution of *vssadmin.exe delete shadows /all /quiet*, as well as terminating specific processes and services indicated in its runtime configuration, some of which are antivirus-related processes and services.

Processes	Services
a2service.exe	acronis vss provider
a2start.exe	acronis vss provider
aawservice.exe	acronisagent
ashserv.exe	acronisagent
avengine.exe	acronisagentd
avkwctl.exe	avbackup
blackd.exe	avbackupd
cfp.exe	ccevtmgr
fsav32.exe	macmnsvc
fsdfwd.exe	macmnsvcd
fsguiexe.exe	masvc
kpf4gui.exe	masvcd
mcods.exe	mcshield
mcpalmcfg.exe	sentinelagent
mcproxy.exe	sentinelagentd
mcregwiz.exe	sentinelhelperservice
mcsacore.exe	sentinelhelperserviced
mcshield.exe	sentinelstaticengine

mpfagent.exe	sentinelstaticengined
mpfservice.exe	shmonitor
msmpeng.exe	shmonitord
msscli.exe	smcinst
nisum.exe	tmccsf
ntrtscan.exe	tmccsfd
pccpfw.exe	tmlisten
tmntsrv.exe	tmlistend

Table 3. Some of the antivirus-related processes and services terminated by Agenda

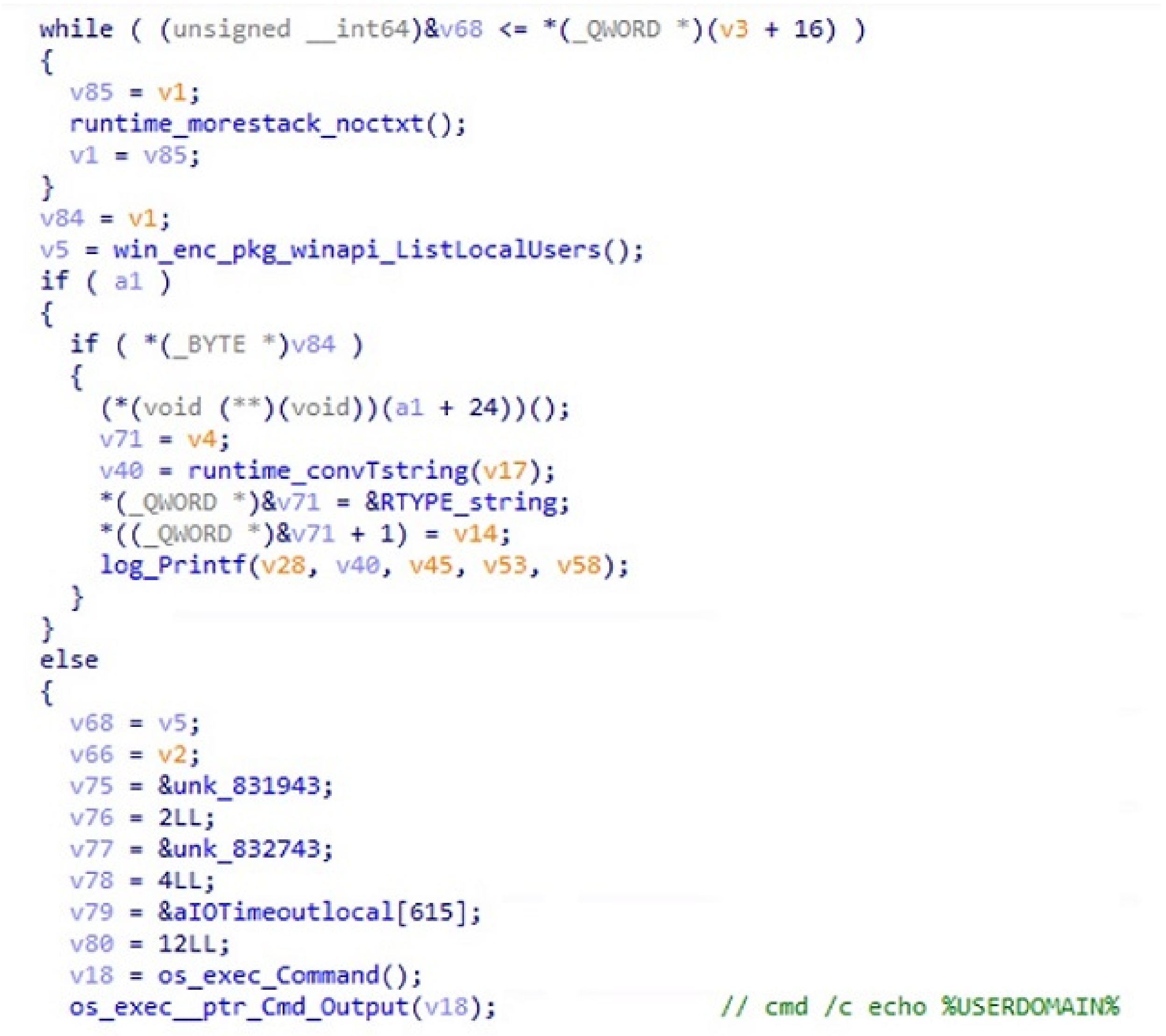
After its initial routine, Agenda proceeds to create the runonce autostart entry *aster pointing to enc.exe, which is a dropped copy of itself under the Public folder:

*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce*aster = %Public%\enc.exe*

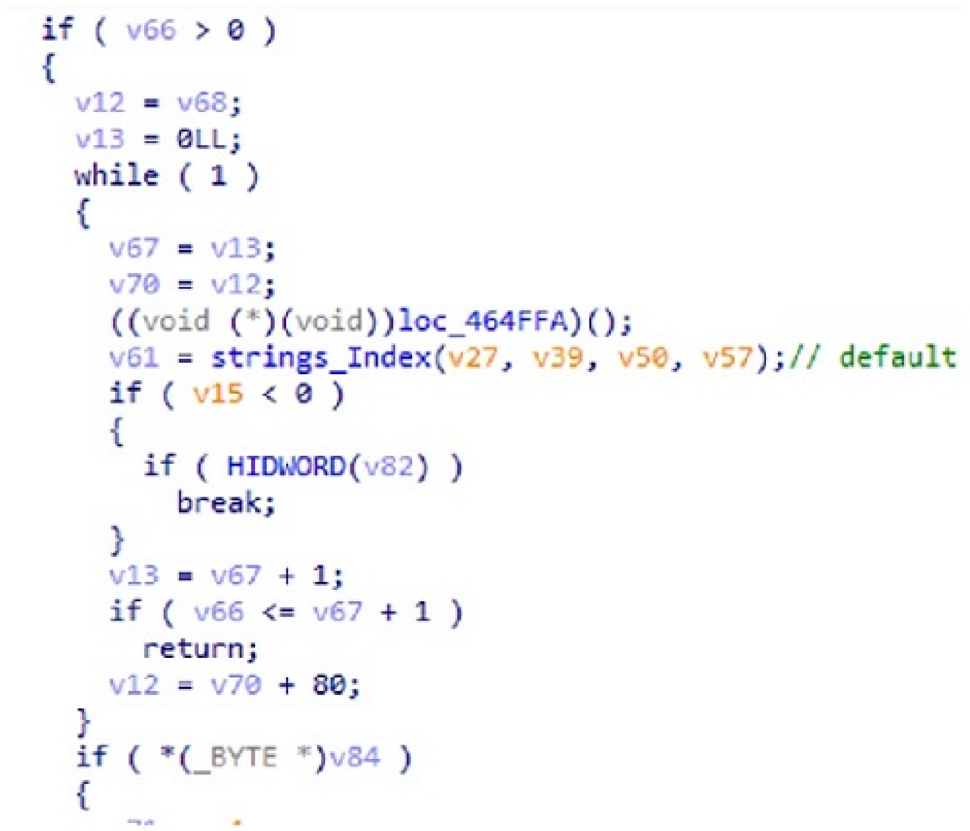
Changing user passwords and rebooting in safe mode

Agenda also deploys a detection evasion technique during encryption: It changes the default user's password and enables automatic login with the new login credentials. This feature can be enabled using the -safe command-line argument. Similar to REvil, Agenda reboots the victim's machine in safe mode and then proceeds with the encryption routine upon reboot.

To begin, Agenda lists all local users found on the device and then checks which one is set as the default user.



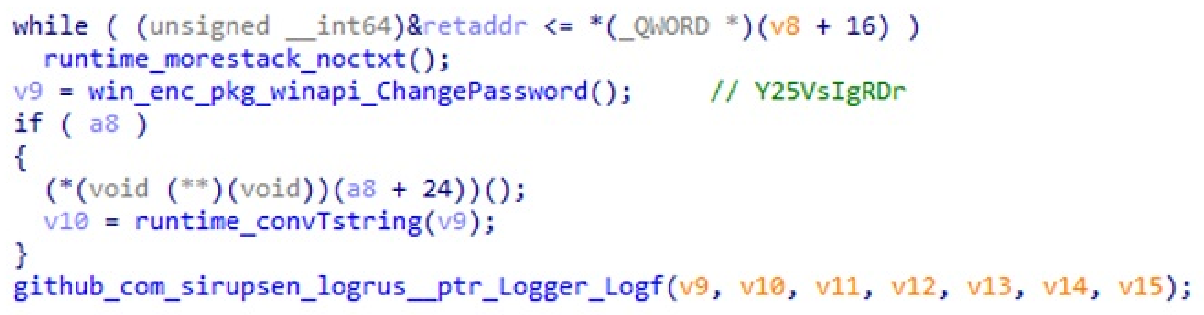
```
while ( (unsigned __int64)&v68 <= *(_QWORD *)(v3 + 16) )
{
  v85 = v1;
  runtime_morestack_noctxt();
  v1 = v85;
}
v84 = v1;
v5 = win_enc_pkg_winapi_ListLocalUsers();
if ( a1 )
{
  if ( *(_BYTE *)v84 )
  {
    (*(void (**)(void))(a1 + 24))();
    v71 = v4;
    v40 = runtime_convTstring(v17);
    *(_QWORD *)&v71 = &RTYPE_string;
    *((_QWORD *)&v71 + 1) = v14;
    log_Printf(v28, v40, v45, v53, v58);
  }
}
else
{
  v68 = v5;
  v66 = v2;
  v75 = &unk_831943;
  v76 = 2LL;
  v77 = &unk_832743;
  v78 = 4LL;
  v79 = &aIOTimeoutlocal[615];
  v80 = 12LL;
  v18 = os_exec_Command();
  os_exec__ptr_Cmd_Output(v18);                  // cmd /c echo %USERDOMAIN%
```



```
  if ( v66 > 0 )
  {
    v12 = v68;
    v13 = 0LL;
    while ( 1 )
    {
      v67 = v13;
      v70 = v12;
      ((void (*)(void))loc_464FFA)();
      v61 = strings_Index(v27, v39, v50, v57);// default
      if ( v15 < 0 )
      {
        if ( HIDWORD(v82) )
          break;
      }
      v13 = v67 + 1;
      if ( v66 <= v67 + 1 )
        return;
      v12 = v70 + 80;
    }
    if ( *(_BYTE *)v84 )
    {
```

Figure 6. The function used by Agenda to determine the default user from local users

Upon finding the default user, Agenda changes the user's password to *Y25VsIgRDr*.



```
while ( (unsigned __int64)&retaddr <= *(_QWORD *)(v8 + 16) )
  runtime_morestack_noctxt();
v9 = win_enc_pkg_winapi_ChangePassword();     // Y25VsIgRDr
if ( a8 )
{
  (*(void (**)(void))(a8 + 24))();
  v10 = runtime_convTstring(v9);
}
github_com_sirupsen_logrus__ptr_Logger_Logf(v9, v10, v11, v12, v13, v14, v15);
```

Figure 7. The function used by Agenda to change the default user's password

It then proceeds to configure the Winlogon registry entry, setting the data to each of these values:

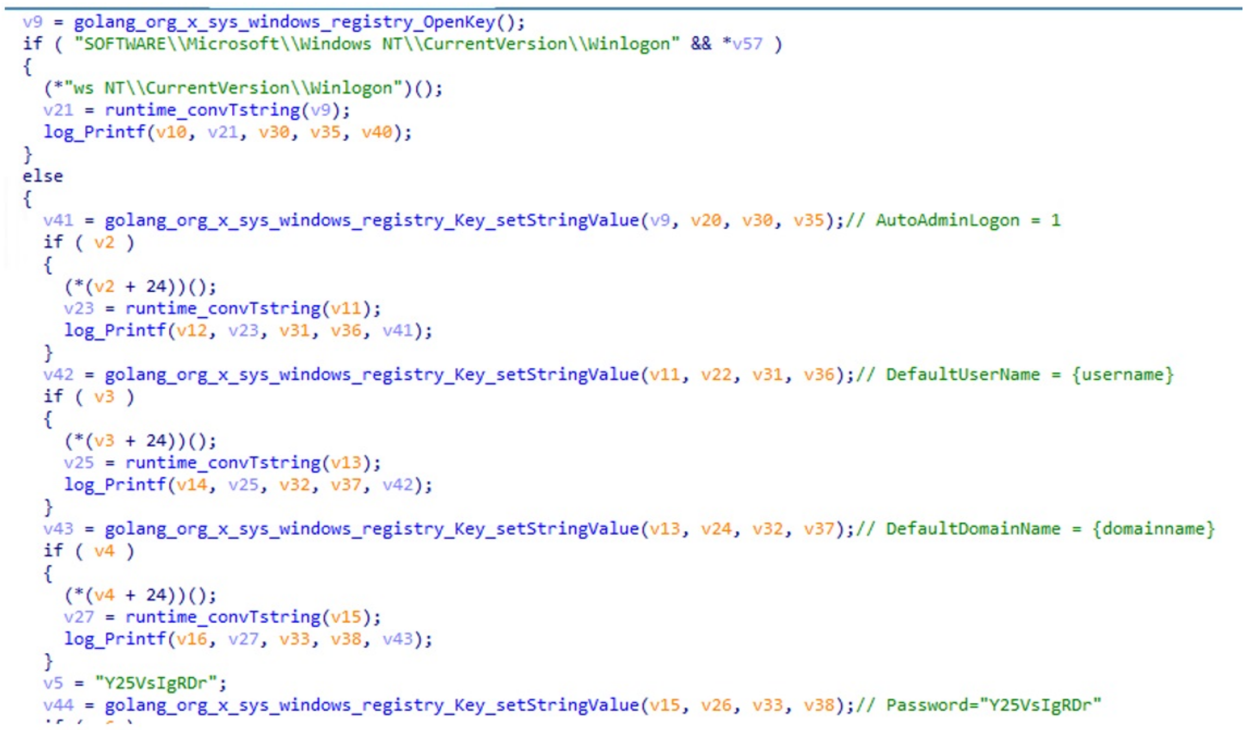
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\

AutoAdminLogon value =1

DefaultUserName = {username}

DefaultDomainName ={domainname}

DefaultPassword={ Y25VsIgRDr}



```
v9 = golang_org_x_sys_windows_registry_OpenKey();
if ( "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon" && *v57 )
{
  (*"ws NT\\CurrentVersion\\Winlogon")();
  v21 = runtime_convTstring(v9);
  log_Printf(v10, v21, v30, v35, v40);
}
else
{
  v41 = golang_org_x_sys_windows_registry_Key_setStringValue(v9, v20, v30, v35);// AutoAdminLogon = 1
  if ( v2 )
  {
    (*(v2 + 24))();
    v23 = runtime_convTstring(v11);
    log_Printf(v12, v23, v31, v36, v41);
  }
  v42 = golang_org_x_sys_windows_registry_Key_setStringValue(v11, v22, v31, v36);// DefaultUserName = {username}
  if ( v3 )
  {
    (*(v3 + 24))();
    v25 = runtime_convTstring(v13);
    log_Printf(v14, v25, v32, v37, v42);
  }
  v43 = golang_org_x_sys_windows_registry_Key_setStringValue(v13, v24, v32, v37);// DefaultDomainName = {domainname}
  if ( v4 )
  {
    (*(v4 + 24))();
    v27 = runtime_convTstring(v15);
    log_Printf(v16, v27, v33, v38, v43);
  }
  v5 = "Y25VsIgRDr";
  v44 = golang_org_x_sys_windows_registry_Key_setStringValue(v15, v26, v33, v38);// Password="Y25VsIgRDr"
```

Figure 8. The Winlogon registry entry configured by Agenda

Upon changing the default user's password and enabling automatic login, Agenda reboots the victim's machine in safe mode via this command:

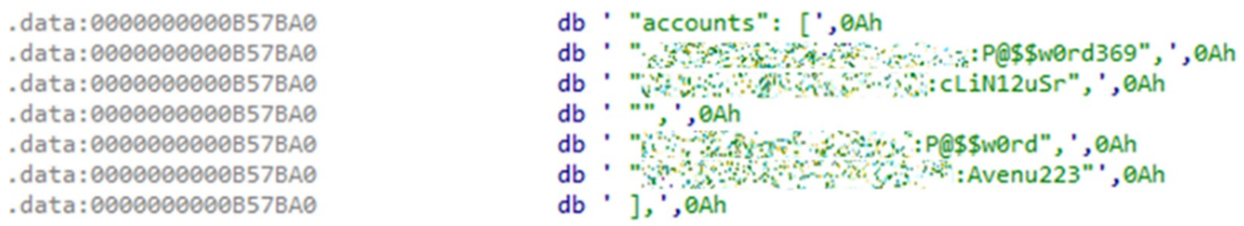
C:\windows\system32\bcdedit.exe /set safeboot{current} network

The ransomware also reboots the machine in normal mode after the encryption using this command:

C:\Windows\System32\bcdedit.exe /set safeboot network bcdedit /deletevalue {default} safeboot

Impersonation of legitimate accounts

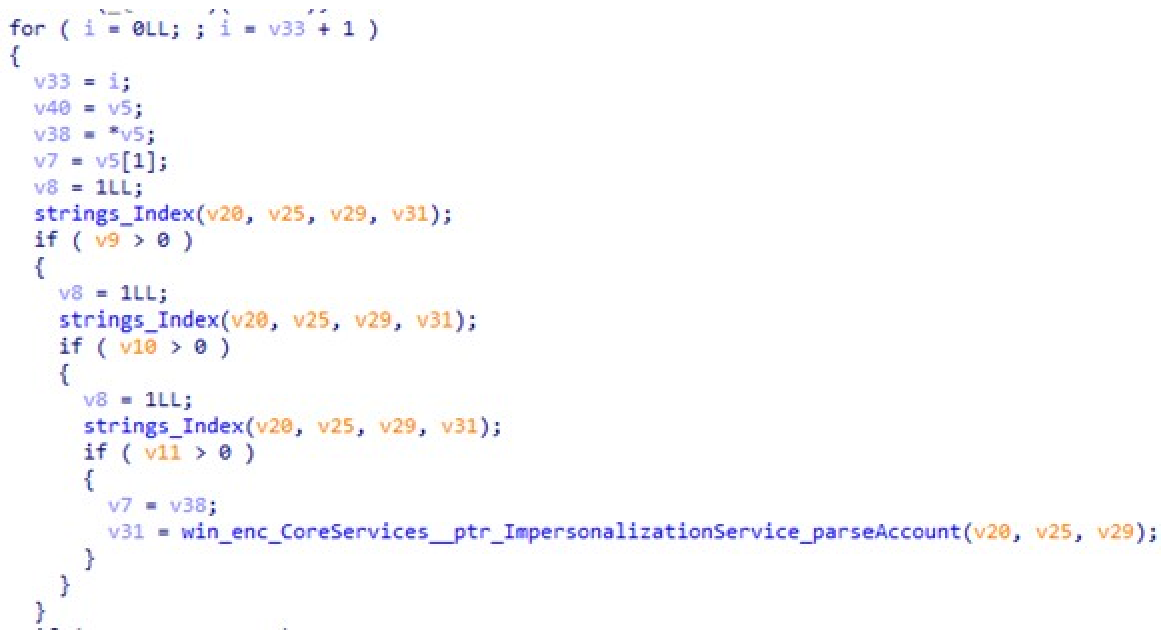
Another feature of Agenda is its ability to abuse local account credentials to execute the ransomware binary, using the embedded login credentials in its runtime configuration.



```
.data:0000000000B57BA0                  db ' "accounts": [',0Ah
.data:0000000000B57BA0                  db ' "[redacted]:P@$$w0rd369",',0Ah
.data:0000000000B57BA0                  db ' "[redacted]:cLiN12uSr",',0Ah
.data:0000000000B57BA0                  db ' "",',0Ah
.data:0000000000B57BA0                  db ' "[redacted]:P@$$w0rd",',0Ah
.data:0000000000B57BA0                  db ' "[redacted]:Avenu223"',0Ah
.data:0000000000B57BA0                  db ' ],',0Ah
```

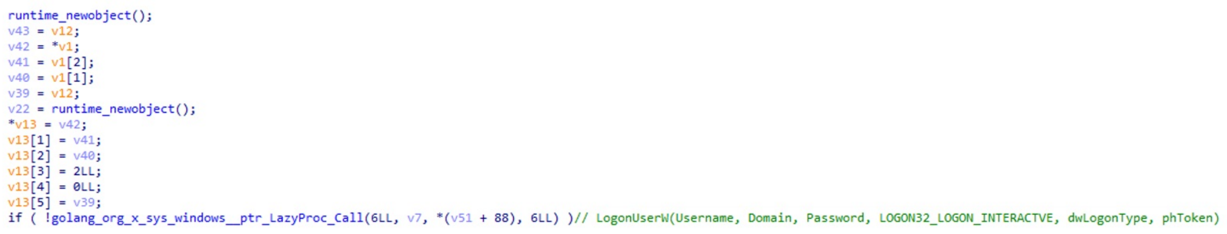
Figure 9. Agenda's embedded local account credentials

Agenda begins the user impersonation by parsing the accounts in the runtime configuration and then separating them into username, domain, and password. It will use this data to attempt logging a user on to the local computer via the API LogonUserW.



```
for ( i = 0LL; ; i = v33 + 1 )
{
  v33 = i;
  v40 = v5;
  v38 = *v5;
  v7 = v5[1];
  v8 = 1LL;
  strings_Index(v20, v25, v29, v31);
  if ( v9 > 0 )
  {
    v8 = 1LL;
    strings_Index(v20, v25, v29, v31);
    if ( v10 > 0 )
    {
      v8 = 1LL;
      strings_Index(v20, v25, v29, v31);
      if ( v11 > 0 )
      {
        v7 = v38;
        v31 = win_enc_CoreServices__ptr_ImpersonalizationService_parseAccount(v20, v25, v29);
      }
    }
  }
```

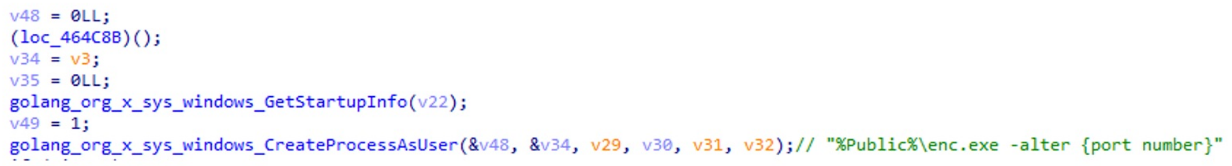
Figure 10. The function used by Agenda to parse the accounts field in the runtime configuration



```
runtime_newobject();
v43 = v12;
v42 = *v1;
v41 = v1[2];
v40 = v1[1];
v39 = v12;
v22 = runtime_newobject();
*v13 = v42;
v13[1] = v41;
v13[2] = v40;
v13[3] = 2LL;
v13[4] = 0LL;
v13[5] = v39;
if ( !golang_org_x_sys_windows__ptr_LazyProc_Call(6LL, v7, *(v51 + 88), 6LL) )// LogonUserW(Username, Domain, Password, LOGON32_LOGON_INTERACTVE, dwLogonType, phToken)
```

Figure 11. Agenda performing logon using a parsed account

Agenda then proceeds to generate a random port number, which it will use in the execution of the ransomware binary through the API CreateProcessAsUserW in conjunction with the command-line argument *-alter*.



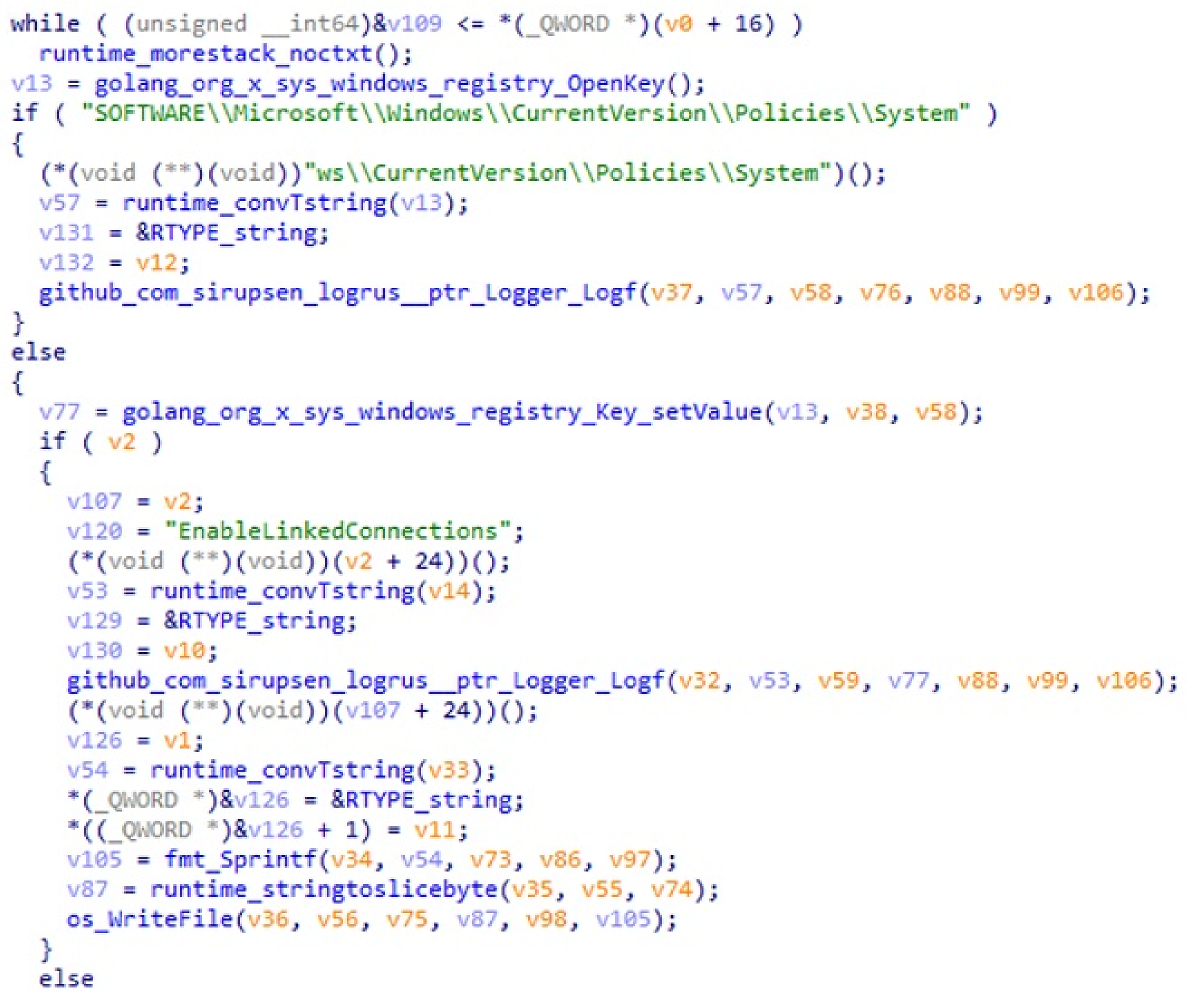
```
v48 = 0LL;
(loc_464C8B)();
v34 = v3;
v35 = 0LL;
golang_org_x_sys_windows_GetStartupInfo(v22);
v49 = 1;
golang_org_x_sys_windows_CreateProcessAsUser(&v48, &v34, v29, v30, v31, v32);// "%Public%\enc.exe -alter {port number}"
```

Figure 12. Agenda creating a new process with the -alter argument

Allowing network sharing

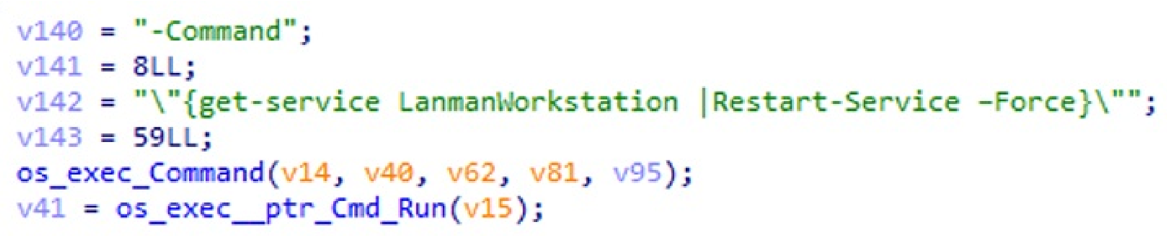
Agenda is also associated with the compromise of an entire network and its shared drivers. It is not only about the encryption of data on one workstation.

The ransomware adds a registry and then restarts the *LanmanWorkstation* service. After adding a new registry, it uses *key [EnableLinkedConnections* = *1]* in the Enabling Mapped Drives drivers and then in restarting the *LanmanWorkstation* service. This will allow Agenda to list network drives in elevated programs like cmd.



```
while ( (unsigned __int64)&v109 <= *(_QWORD *)(v0 + 16) )
  runtime_morestack_noctxt();
v13 = golang_org_x_sys_windows_registry_OpenKey();
if ( "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System" )
{
  (*(void (**)(void))"ws\\CurrentVersion\\Policies\\System")();
  v57 = runtime_convTstring(v13);
  v131 = &RTYPE_string;
  v132 = v12;
  github_com_sirupsen_logrus__ptr_Logger_Logf(v37, v57, v58, v76, v88, v99, v106);
}
else
{
  v77 = golang_org_x_sys_windows_registry_Key_setValue(v13, v38, v58);
  if ( v2 )
  {
    v107 = v2;
    v120 = "EnableLinkedConnections";
    (*(void (**)(void))(v2 + 24))();
    v53 = runtime_convTstring(v14);
    v129 = &RTYPE_string;
    v130 = v10;
    github_com_sirupsen_logrus__ptr_Logger_Logf(v32, v53, v59, v77, v88, v99, v106);
    (*(void (**)(void))(v107 + 24))();
    v126 = v1;
    v54 = runtime_convTstring(v33);
    *(_QWORD *)&v126 = &RTYPE_string;
    *((_QWORD *)&v126 + 1) = v11;
    v105 = fmt_Sprintf(v34, v54, v73, v86, v97);
    v87 = runtime_stringtoslicebyte(v35, v55, v74);
    os_WriteFile(v36, v56, v75, v87, v98, v105);
  }
  else
```

Figure 13. Agenda changing the registry value of EnableLinkedConnection to 1

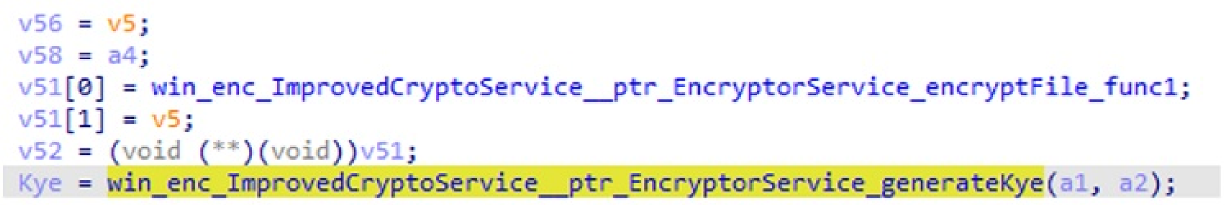


```
v140 = "-Command";
v141 = 8LL;
v142 = "\"{get-service LanmanWorkstation |Restart-Service -Force}\"";
v143 = 59LL;
os_exec_Command(v14, v40, v62, v81, v95);
v41 = os_exec__ptr_Cmd_Run(v15);
```

Figure 14. Agenda restarting the LanmanWorkstation service

Encryption algorithm

Agenda uses AES-256 for encrypting files and RSA-2048 for encrypting the generated key. To do so, it first generates the key and initialization vector (IV) that it will use for encryption by using the function *generateKye*, and then uses the API *rand_read()*.

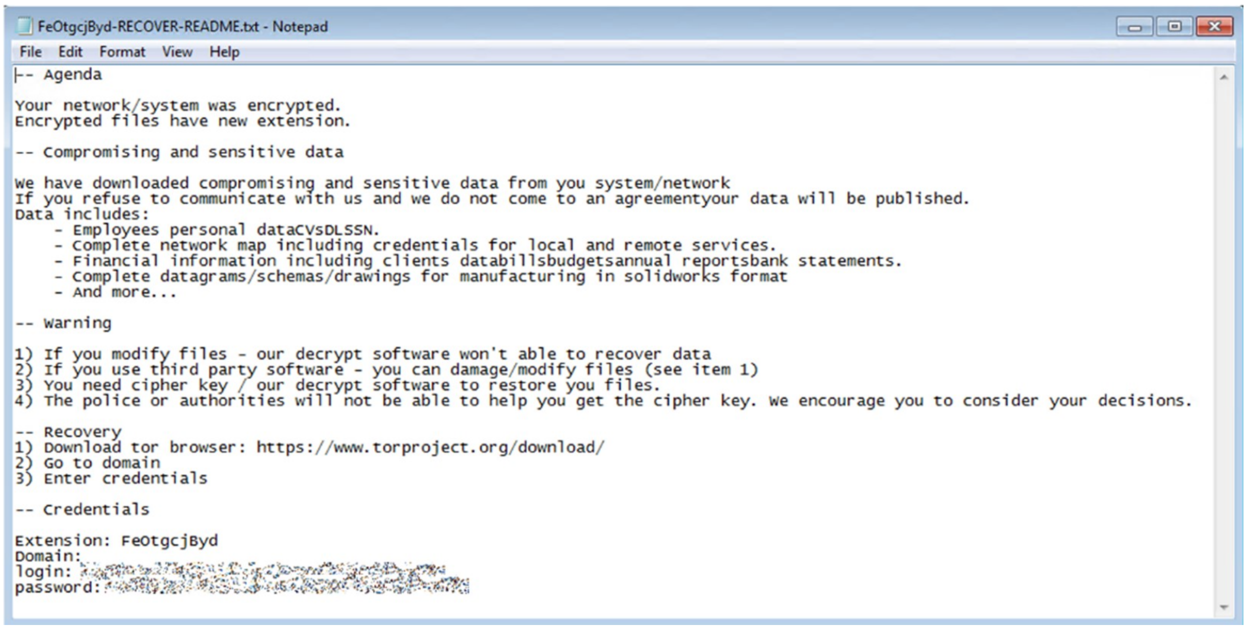


```
v56 = v5;
v58 = a4;
v51[0] = win_enc_ImprovedCryptoService__ptr_EncryptorService_encryptFile_func1;
v51[1] = v5;
v52 = (void (**)(void))v51;
Kye = win_enc_ImprovedCryptoService__ptr_EncryptorService_generateKye(a1, a2);
```

Figure 15. The function used by Agenda to generate a random key

With this randomly generated key, Agenda proceeds to use AES-256 to encrypt target files. Lastly, it encrypts the key using RSA-2048 through the embedded public key from the runtime configuration.

After successful encryption, Agenda renames the encrypted files by appending the company ID indicated in the runtime configuration. It then drops the ransom note *{company_id}-RECOVER-README.txt* in each encrypted directory.



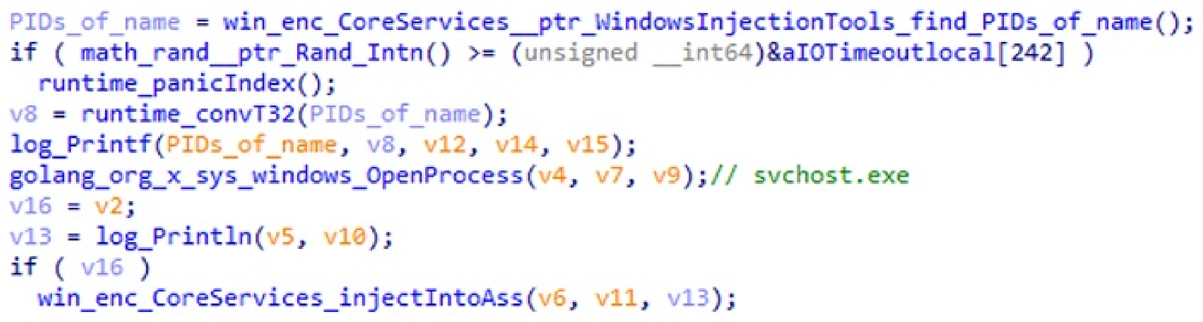
FeOtgcjByd-RECOVER-README.txt - Notepad
File Edit Format View Help

```
|-- Agenda

Your network/system was encrypted.
Encrypted files have new extension.

-- Compromising and sensitive data

We have downloaded compromising and sensitive data from you system/network
If you refuse to communicate with us and we do not come to an agreementyour data will be published.
Data includes:
    - Employees personal dataCVsDLSSN.
    - Complete network map including credentials for local and remote services.
    - Financial information including clients databillsbudgetsannual reportsbank statements.
    - Complete datagrams/schemas/drawings for manufacturing in solidworks format
    - And more...

-- Warning

1) If you modify files - our decrypt software won't able to recover data
2) If you use third party software - you can damage/modify files (see item 1)
3) You need cipher key / our decrypt software to restore you files.
4) The police or authorities will not be able to help you get the cipher key. We encourage you to consider your decisions.

-- Recovery
1) Download tor browser: https://www.torproject.org/download/
2) Go to domain
3) Enter credentials

-- Credentials

Extension: FeOtgcjByd
Domain:
login: [redacted]
password:
```

Figure 16. Agenda's ransom note

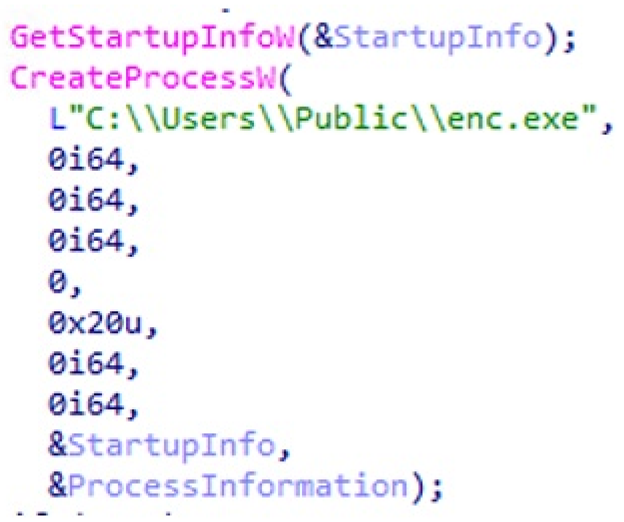
Process injection

Agenda drops *pwndll.dll*, detected as *Trojan.Win64.AGENDA.SVT*, in the Public folder. The file *pwndll.dll* is a patched DLL from the legitimate DLL WICloader.dll written in C, not Go. Agenda injects this DLL into *svchost.exe* to allow continuous execution of the ransomware binary.



```
PIDs_of_name = win_enc_CoreServices__ptr_WindowsInjectionTools_find_PIDs_of_name();
if ( math_rand__ptr_Rand_Intn() >= (unsigned __int64)&aIOTimeoutlocal[242] )
  runtime_panicIndex();
v8 = runtime_convT32(PIDs_of_name);
log_Printf(PIDs_of_name, v8, v12, v14, v15);
golang_org_x_sys_windows_OpenProcess(v4, v7, v9);// svchost.exe
v16 = v2;
v13 = log_Println(v5, v10);
if ( v16 )
  win_enc_CoreServices_injectIntoAss(v6, v11, v13);
```

Figure 17. Agenda injecting pwndll.dll into svchost.exe



```
GetStartupInfoW(&StartupInfo);
CreateProcessW(
  L"C:\\Users\\Public\\enc.exe",
  0i64,
  0i64,
  0i64,
  0,
  0x20u,
  0i64,
  0i64,
  &StartupInfo,
  &ProcessInformation);
```

Figure 18. Agenda using pwndll.dll to execute the ransomware sample

Conclusion and solutions

Ransomware continues to evolve, developing more sophisticated methods and techniques to trap organizations. Our investigation shows how the new targeted ransomware Agenda is written in the Go language, making it harder to detect and analyze.

This ransomware has techniques for evading detection by taking advantage of the “safe mode” feature of a device to proceed with its encryption routine unnoticed. The ransomware also takes advantage of local accounts to log on as spoofed users and execute the ransomware binary, further encrypting other machines if the logon attempt is successful. It also terminates numerous processes and services, and ensures persistence by injecting a DLL into svchost.exe.

End users and organizations alike can mitigate the risk of infection from ransomware like Agenda by following these security best practices:

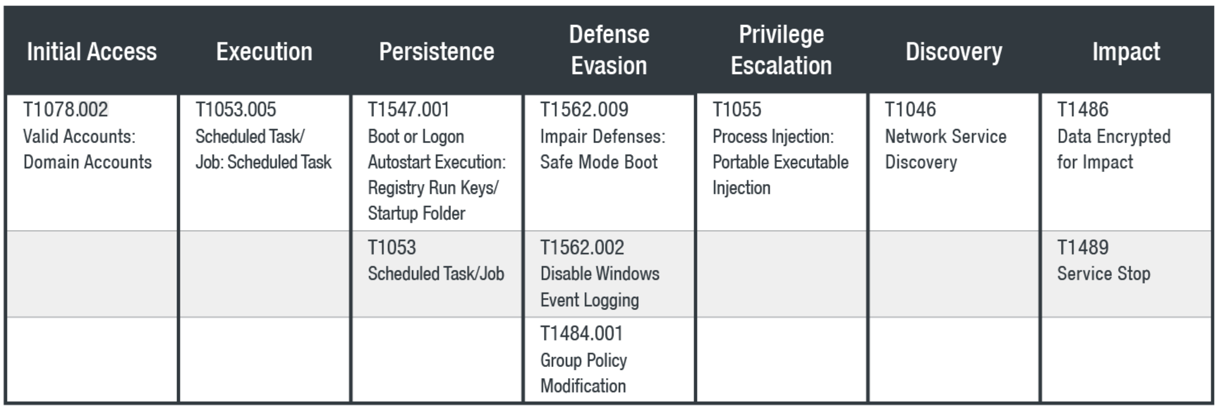
- Enable multifactor authentication (MFA) to prevent attackers from performing lateral movement inside a network.
- Adhere to the 3-2-1 rule when backing up important files. This involves creating three backup copies on two different file formats, with one of the copies stored in a separate location.
- Patch and update systems regularly. It’s important to keep operating systems and applications up to date, preventing malicious actors from exploiting any software vulnerabilities.

Organizations can also benefit from security solutions that offer multilayered detection and response, such as Trend Micro Vision One™, which has multilayered protection and behavior detection capabilities that help block suspicious behavior and tools before ransomware can do any damage. Trend Micro Apex One™ also provides next-level automated threat detection and response to protect endpoints against advanced issues, like fileless threats and ransomware.

Additional insights provided by Eleazar Valles and Sherif Magdy.

For more information about the indicators of compromise, download this document.

MITRE ATT&CK tactics and techniques



Initial Access	Execution	Persistence	Defense Evasion	Privilege Escalation	Discovery	Impact
T1078.002 Valid Accounts: Domain Accounts	T1053.005 Scheduled Task/ Job: Scheduled Task	T1547.001 Boot or Logon Autostart Execution: Registry Run Keys/ Startup Folder	T1562.009 Impair Defenses: Safe Mode Boot	T1055 Process Injection: Portable Executable Injection	T1046 Network Service Discovery	T1486 Data Encrypted for Impact
		T1053 Scheduled Task/Job	T1562.002 Disable Windows Event Logging			T1489 Service Stop
			T1484.001 Group Policy Modification			